

Article

An Adaptive Agent-Based Model of Homing Pigeons: A Genetic Algorithm Approach

Francis Oloo * and Gudrun Wallentin

Department of Geoinformatics (Z_GIS), University of Salzburg, Schillerstraße 30, 5020 Salzburg, Austria; gudrun.wallentin@sbg.ac.at

* Correspondence: francisomondi.oloo@sbg.ac.at; Tel.: +43-662-8044-7569

Academic Editor: Wolfgang Kainz

Received: 14 October 2016; Accepted: 16 January 2017; Published: 21 January 2017

Abstract: Conventionally, agent-based modelling approaches start from a conceptual model capturing the theoretical understanding of the systems of interest. Simulation outcomes are then used “at the end” to validate the conceptual understanding. In today’s data rich era, there are suggestions that models should be data-driven. Data-driven workflows are common in mathematical models. However, their application to agent-based models is still in its infancy. Integration of real-time sensor data into modelling workflows opens up the possibility of comparing simulations against real data during the model run. Calibration and validation procedures thus become automated processes that are iteratively executed during the simulation. We hypothesize that incorporation of real-time sensor data into agent-based models improves the predictive ability of such models. In particular, that such integration results in increasingly well calibrated model parameters and rule sets. In this contribution, we explore this question by implementing a flocking model that evolves in real-time. Specifically, we use genetic algorithms approach to simulate representative parameters to describe flight routes of homing pigeons. The navigation parameters of pigeons are simulated and dynamically evaluated against emulated GPS sensor data streams and optimised based on the fitness of candidate parameters. As a result, the model was able to accurately simulate the relative-turn angles and step-distance of homing pigeons. Further, the optimised parameters could replicate loops, which are common patterns in flight tracks of homing pigeons. Finally, the use of genetic algorithms in this study allowed for a simultaneous data-driven optimization and sensitivity analysis.

Keywords: adaptive rulesets; genetic algorithms; sensors; agent-based modelling; optimization

1. Introduction

Traditionally, agent-based models (ABM) have been formulated from a hypothesis (or rules of behaviour) of the systems of interest [1]. Empirical data are introduced at later stages to facilitate calibration and validation of the model in an iterative process of exploration [2]. The result is a set of state variables that produce a realistic representation of processes and behavioural mechanisms of the underlying system and therefore can be used to check the validity of the presumed mechanism of the system. An impediment to the use of ABM in investigating spatio-temporal systems has been the lack of fine-scale data [3] to describe the behaviour of individual agents and to be used for calibration, verification, and validation of such models. Recent improvements in data collection and transmission, particularly due to the advent of sensor technologies, present an opportunity for the incorporation of the rich data emanating from such platforms into modelling and simulation environments [4].

Emergence of miniaturized sensors and intelligent sensor networks [5] provides an opportunity of deploying sensors in remote environments and using the resulting sensor data streams to capture dynamic local characteristics of individual drivers of geographic processes. These data can be used to monitor, investigate, and to understand the influence of such individual behaviours on the overall

system level outcomes. Additionally, wireless sensor networks can facilitate transparent and efficient transfer of sensor resources, hence reducing turnaround time between data collection and analysis, and visualization of the results. The ability of sensors, particularly those deployed under the “breed” of wireless sensor networks (WSN) [6], to dynamically capture spatio-temporal characteristics of the systems at hand and to transfer the information in real-time also raises the question on the suitability of traditional methods in model specification [7] and geospatial analysis. One suggestion to efficient utilization of sensor resources has been to shift from centralized, desktop-based applications towards a distributed web-based geospatial services [5,6]. This has led to extensive interest and research with an aim of developing open standards to facilitate a standardized means of implementation, documentation, discovery, and access of sensor oriented services [7,8]. Consequently, sensor networks have been applied to a great variety of domains, including environmental [4,9,10], public health [11,12], hydrology [13], and mobility [14]. In the same context, scientists in the field of animal movement ecology have developed standardized online animal movement data repositories such as Movebank [15,16], Wireless Remote Animal Monitoring (WRAM) database [17], and Information System for Analysis and Management of Ungulate Data (ISAMUD) [18] among others [19]. As a result, sensor observations have become an integral part of research on movement behaviours [14,20,21].

These initial successes in the implementation of sensor and related data management technologies have led to an interest in dynamic incorporation of data into simulation models. The most prominent attempt at dynamic assimilation of sensor data into simulation models was made through the framework of Dynamic Data-Driven Application Simulation (DDDAS) [22–26]. An implementation of DDDAS methods in agent-based modelling environment was investigated by [27] who executed a bi-directional communication between a camera system and a model of animal movements. In an attempt to address the challenges related to data management and dynamic visualization of the results of dynamic data-driven ABMs, parallel simulation under the framework of Distributed Dynamic Data-driven Simulation and Analysis System (4D-SAS) has been proposed [28]. Furthermore, there is evidence that behaviour patterns of agents as captured by sensor observations can facilitate and improve data assimilation in dynamic data driven simulation models [29]. Whereas dynamic-data assimilation has largely been successful in mathematical models [30,31], there have only been limited attempts to implement data-driven ABMs [32]. In a related study, emulated sensor data streams were used in a model-sensor framework to evaluate agent decisions against “sensor” data during the model runs [33]. An outstanding aspect of research that is yet to be explored conclusively is on how dynamic data can be used to improve specification of the behaviour of agents in rule-based and spatially explicit ABMs.

To dynamically improve the spatial and temporal behaviour of agents, it is important to consider how such agents adapt to variations in the characteristics of their environment. Adaptation is an important attribute of ABMs and relies on the ability of agents to sense their environments, to learn about possible actions to take in different circumstances, and respond to stimuli from other agents or from their environments [34]. Learning within ABMs has been achieved through methods of reinforcement learning [35], evolutionary (genetic) algorithms [36,37] and machine learning [38]. Spatial learning in particular is the precursor to successful adaptation [39,40] and has been identified as one of the challenges to representation of spatially aware agents [35]. Reinforcement learning is particularly useful in cases where the environment is represented in simple, discrete, and deterministic states [41]. Machine learning has been used in agent-based models for data mining to recognize patterns in data [42]. Genetic algorithms are best suited for ABMs where learning and adaptation are integral attributes of the agents in the system of interest. Moreover, genetic algorithms have been suggested as suitable heuristic algorithms for various aspects of ABM design including; directed parameter search [43], calibration [44], parameter optimization, and sensitivity analysis [45].

Conceptually, genetic algorithms are computer-based techniques that borrow from rules of natural selection to solve difficult optimisation problems [37]. The basic concepts of genetic algorithms and their framework for adaptation were initially defined by John Holland in the 1960s [46,47]. As in natural

selection, candidate solutions to a problem are coded as “chromosomes” [48]. The algorithms then evolve populations of candidate solutions to discrete problems by using tools of natural selection [38]. In the simplest form, evolution is achieved by applying three genetic operators of selection, mutation, and crossover (recombination) [48,49]. Selection operators use a measure of fitness to select a set of chromosomes in a population as candidates for reproduction. The choice of a realistic fitness function to measure how well candidate chromosomes simulate behaviours evident in the real world remains an outstanding challenge in implementing genetic algorithms [50]. Mutation operators on the other hand, randomly flip or vary a gene/locus within the chromosome and can either have negative or positive influence on the performance of the resulting offspring. Crossover or recombination operators swap portions of candidate chromosomes from the parents involved in reproduction in order to create new sets of chromosomes of the offspring. Apart from the three basic operators of selection, mutation and crossover, a replacement operator specifies the mechanism of evolution of generations within a population. In a generational genetic algorithm, all parent solutions are replaced at the end of each generation while in steady-state/overlapping algorithms only a percentage (which is usually less than 100%) of the parents are eliminated in each generation and replaced by offspring bred during reproduction [37]. Genetic algorithms have been applied to agent-based models of spatially explicit systems with most examples being in the area of ecology [36,51,52]. In the documented cases, their application has been in optimization or to tune model parameters [50,53] and to identify behaviours of interest from parameter space [43]. However, the question remains, how to combine fine-grained sensor observations and evolutionary methods to enhance ABM specification, calibration, and validation in a data-driven approach.

In this contribution, we hypothesize that dynamic integration of sensor data into spatially explicit ABMs improves the specification of such models. Additionally, we believe that using sensor data to dynamically evaluate and to optimise model parameters improves the predictive ability of the models. In particular, sensor observations about the local level spatial behaviour of agents are critical for understanding spatio-temporal heterogeneity of different phenomena and processes. We reason that a combination of dynamic sensor data streams and evolutionary algorithms can provide the necessary components for model specification, verification, and automated calibration. We therefore used Global Positioning System (GPS) tracks of pigeons from a real experiment to emulate dynamic sensor data streams. The dynamic sensor data streams were fed into an ABM used to dynamically evaluate navigation decisions of simulated homing pigeons on the fly. Additionally, we used genetic algorithms to evolve navigation and flocking parameters of homing pigeons. This was intended to identify an optimal range of parameters that can be used to reproduce realistic navigation paths of pigeon agents from a release site to a home loft.

Related Work

Data-driven ABMs that are related to the one that we present here have been the subject of investigation by a number of researchers. Real-time measurements have been injected dynamically into fire simulation models to improve the specification of such models [23,54,55]. In the same line, dynamic-data driven genetic algorithms have been used to automatically adjust input values of fire simulators in order to take advantage of real fire behaviour [56]. Other studies have used real-time data to augment crowd simulation models [57]. In particular, behaviour detection models have been used to facilitate data assimilation in ABMs of smart environments [29]. Attempts to improve management and visualization of results of models that rely on dynamic spatial data are also of note [28]. Within animal movement ecology, navigation and flocking behaviour of pigeons have been found to govern the routing decisions of pigeons on their homing flights [58]. Additionally, GPS experiments have revealed that pigeons use topography-induced detours and cognitive navigation maps to find their targets [59].

This paper is organized as follows: the types of data that we used to design the model and the details of the parameters, procedures and sub-models as specified in the model are presented in

Section 2. The results from the model specification and those of the data analysis are presented in Section 3. In Section 4, we provide a discussion of the results and briefly outline what we consider to be the next frontier for dynamic sensor data integration into agent-based models. In the final section, we provide a conclusion to summarize on the achievements, challenges, and future research directions that can build on this study.

2. Materials and Methods

2.1. Data

GPS tracks containing flight details of homing pigeons (*Columba livia*) provided the main dataset in this study. The data were sourced from an experimental research on leadership among homing pigeons in Seuzach, Switzerland [60,61]. We accessed the data from Movebank, which is an online platform to report and share animal tracking data (www.movebank.org) [15,16]. Specifically, the data contained information on five separate homing flights. In each flight, tracks of a flock containing at least seven pigeons were captured. GPS coordinates, speed, flight altitude, and tag identifiers of each bird were recorded at a time interval of 0.25 s during the experimental flights. A digital elevation model (DEM) of the area of research with a 25 m spatial resolution was also used in the study as will be explained in the subsequent sections.

2.2. The Model

We designed a flocking and navigation model of the homing pigeons to simulate the flight of pigeon flocks from a release site to the home loft. We followed “ODD” (Overview, Design concepts, and Details) protocol [62] in model description and implemented the model in Netlogo software [63]. The ODD protocol aims at simplifying the process of writing and reading model descriptions and has been broadly embraced within the agent-based modelling community [64,65]. The specific aspects of the ODD protocol that we captured in our model are as outlined in the subsequent sections.

2.2.1. Purpose

The purpose of the model was to gain a better understanding of the navigation behaviour of homing pigeons by identifying an optimal range of parameters that govern their navigation decisions.

2.2.2. Entities, State Variables and Scales

The entities represented in the model include pigeon agents, homing loft, and the environment. Two sets of pigeon agents are represented in the model; “real” agents that derive their navigation rules from empirical data as captured in the emulated GPS tracks and “simulated” birds that use flocking and navigation rules to navigate to the homing loft. The number of pigeon agents (“real” or “simulated”) represented in the model ranged from 7 to 9 depending on the number of pigeons captured in the respective empirical flight tracks. A single homing loft is specified in this model and is represented as a stationary agent. The environment is represented as a topographic surface as derived from the DEM. The spatial extent of the area of study is approximately 17 km by 17 km. A single time step in the model represents 0.25 s in the real world. The temporal scale of 0.25 s was chosen because it is the same interval at which the GPS tracks were archived. An entire flight from the release site to the homing loft is approximately 20 min which is approximately 4800 time steps in the model. In order to conclusively capture the state variables that are necessary for this kind of model, we relied on the literature on ABMs for animal movement [66]. The state variables of the pigeon agents include their coordinates, current heading, number of flock mates, step-distance which is variable in each time step, relative turn-angle (in degrees) in the latest time step, sinuosity of the current flight path, and the cumulative flight distance from the release site to the current location. The state variables of the home loft are its coordinates which remain constant throughout the simulation. The state variables of the environment are the coordinates of patches and their associated elevation.

2.2.3. Process Overview and Scheduling

A summarised workflow of the key processes that are implemented in the model are shown in Figure 1; the main processes include initialization of model entities, execution of navigation and flocking procedures, and the evolution of model parameters through genetic algorithms. Navigation behaviours include a map turn and a compass turn. The map turn procedure ensures that the flight routes of pigeon agents are guided by the landscape forms. This behaviour is implemented so that pigeon agents fly along a regular topographic contour. Compass turn borrows from the assumption that pigeons can generally sense the bearing towards their home loft. The compass turn procedure ensures that agents fly towards a general bearing towards their home loft. The flocking behaviour is controlled by separation, alignment and coherence rules as originally defined in the boids model [67]. Genetic algorithm is used to evolve the navigation and the flocking parameters after each time step.

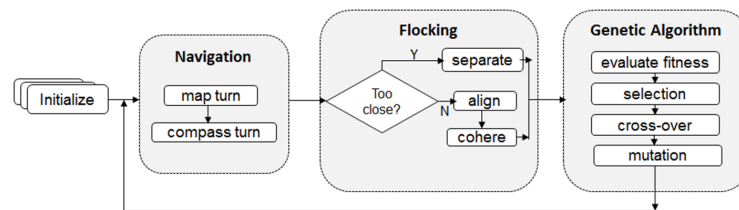


Figure 1. Flow diagram of critical steps in the model including initialization, navigation, flocking, and genetic algorithm implementation within the model.

During the training runs of the model, “real” and “simulated” agents are initialized and simultaneously simulated in the model. In general, simulation encompassed the following processes.

- State variables of pigeon agents are updated (or initialized if at the beginning of the model).
- Agents execute navigation decisions; specifically, “real” agents use values of relative turn angle (*mean-turn*) and *step-distance* which are attributes that are captured in the empirical GPS data streams to orient their heading by an angle equivalent to the *mean-turn* and to move forward by a distance equivalent to the *step-distance*. On the other hand, “simulated” agents choose their navigation behaviours on the basis of two sub-models, which are specified as *map-turn* and *compass-turn*. The aim of the *map-turn* parameter, alternatively referred to as the *elevation-turn* is to re-orient agent headings and allow agents to fly along a controlled topographic contour. On the other hand, *compass-turn* ensures that an agent dynamically maintains a general bearing towards a known home loft during navigation.
- Agents find flock mates with whom to navigate; flocking rules are adopted from Reynolds flocking model [67] and are guided by separation, alignment, and coherence procedures. These procedures are controlled by model parameters which are specified as minimum separation distance, maximum separation turn, maximum alignment turn, and maximum cohere turn. The separation procedure supersedes the other procedures under the flocking behaviour and by it, an agent changes its direction of flight to avoid colliding with nearby flock mates. An implementation of the flocking model is included as one of the library models within NetLogo package [68] and this is what we modified and adopted to fit the specifications and objectives of the model presented herein.
- Apart from the navigation and flocking behaviours, agents can also turn randomly to their left (*turn-random-left*) or to their right (*turn-random-right*) based on a probability (*random-turn-prob*).
- A genetic algorithm is implemented to evolve the initial population of candidate parameters and to optimize the range of flight parameters.
- Output is produced; this includes coordinates of agents, *step-distances*, cumulative turn in the respective time step, sinuosity of the flight path, chromosome of the current agent, and the fitness value associated with the current chromosome of the agent.

2.2.4. Design Concepts

- **Emergence:** We are interested in a range of parameters that reproduce realistic flight paths and observable navigation behaviours of homing pigeon agents. Specifically, we are looking for observable corridors and possible loops in the flight paths that emerge from navigation, flocking, and random decisions of the pigeons.
- **Adaptation:** Agents make adaptive decisions during flocking as well as in identifying optimal flight directions by considering the limits of navigation and flocking parameters.
- **Objectives:** The goal of “simulated” agents is to successfully navigate to the homing loft by following efficient tracks. This is achieved by avoiding areas with abrupt changes in elevation and preferably by navigating in flocks.
- **Sensing:** Pigeon agents can sense other agents (flock mates) in their neighbourhoods. An agent neighbourhood is specified by visible distance (*vision-distance*) and a view angle (*vision-angle*). Additionally, agents can perceive the differences in elevation between their current locations and the surrounding patches in their environments.
- **Collectives:** Pigeon agents prefer to navigate in flocks, which is a social group of pigeon agents.
- **Observation:** Apart from the flight paths of agents that are plotted during simulation, additional charts are plotted to show the variation in mean-turn angles, average step-distance, fitness of candidate chromosomes, and sinuosity of flight paths. In addition, a monitor is used to report the cumulative travel time of agents. The flight time (in minutes) is as shown in Equation (1).

$$t = 0.25 \left(\frac{\text{ticks}}{60} \right) \quad (1)$$

2.2.5. Input Data

Emulated sensor data streams, which were derived from experimental GPS tracks of homing pigeons, provided the navigation parameters for “real” agents. These data were dynamically loaded into the NetLogo modelling environment as GIS point vector data in shapefile format. Table 1 represents the number of data points and the number of sample birds that were captured in each of the five homing flights in the empirical (experimental) data. Apart from the GPS tracks, a DEM was used in the model to represent the spatial variability in the elevation of the navigation environment.

Table 1. Number of data points and number of birds in each empirical homing flight.

Homing Flight	Number of Data Points	Number of Birds
1	4667	8
2	2788	8
3	3500	7
4	3496	8
5	3844	9

2.2.6. Sub-Models

We used a number of sub-models to implement the main processes in the model. Table 2 outlines a list of the sub-models that are specified and implemented in the model. The sub-models of the flocking processes were modified from the NetLogo flocking model [68]. Similarly, we adopted and modified ideas for the *create-new-generation* and *mutate* sub-models from Robby the Robot model in NetLogo [69]. Apart from the specified sub-models (Table 2), we also defined a number of reporter procedures (functions) to generically and iteratively compute different parameters that are important for successful implementation of different model processes and sub-models. These included reporters to calculate state variables such as sinuosity, elevation heading, compass heading and agent fitness among others.

Table 2. Individual sub-models and their purposes.

Sub-Model	Purpose
<i>read-GPS-sensor?</i>	Uses emulated GPS points to create and to guide navigation of “real” agents.
<i>create-birds-agents</i>	Uses the first set of coordinates in the emulated GPS tracks to create “simulated” pigeon agents. Additionally, sets the initial state variables of simulated agents.
<i>display-elevation</i>	Imports the DEM, transforms its coordinates to the model coordinate system, and specifies how the environment is displayed.
<i>map-turn</i>	Uses a defined elevation turn (<i>max-elevation-turn</i>) and the variation of elevation in the locality of an agent to re-orient the heading of the agent.
<i>compass-turn</i>	An agent uses trigonometric functions to determine the bearing to the home loft. The result is compared to the allowable compass turn (<i>max-loft-turn</i>), if the bearing is less than <i>max-loft-turn</i> then the agent reorients its heading to face the direction of the home loft otherwise the agent turns by an angle that is equivalent to the <i>max-loft-turn</i> in the direction of the home loft.
<i>encode-chromosome</i>	Uses a predefined set of model parameters to encode a vector of candidate chromosomes; stochasticity is introduced in the chromosomes by using a normal distribution with a mean of 0.0 and standard deviation of 0.01 to randomly vary the individual chromosome parameters.
<i>create-new-generation</i>	Implements genetic algorithm operators of selection, crossover, mutation, and replacement.
<i>mutate</i>	Randomly selects a gene within the candidate (intermediate) chromosome and uses a normal distribution with a mean of 0.0 and standard deviation of 0.1 to vary the selected gene.
<i>export-results</i>	Generates a tabular output of the agent properties at each time step.

2.2.7. Model Parameters

Model parameters define the limits of capabilities of agents that are represented in a model. Table 3 represents a list of model parameters that were specified in the model. Whereas most of the initial parameters can be adjusted by user, *min-step-distance* and *max-step-distance* were mathematically defined within the model. Specifically, *min-step-distance* is defined from random normal function (*random-normal*) with a mean of 2.0 and standard deviation of 0.5 while *max-step-distance* is calculated as the sum of *min-step-distance* and random number generated from a normal distribution with a mean of 5.0 and standard deviation of 1.0. The values of mean and standard deviation that were used estimate *min-step-distance* and *max-step-distance* variables were pre-calculated in the observed trajectory data.

Table 3. Model parameters.

Parameter	Description	Base Values
<i>max-align-turn</i>	Maximum turning angle that an agent can execute to align to flight direction of flock mates.	8.0
<i>max-cohere-turn</i>	Maximum turning angle that an agent can implement to be in coherence with its flock mates.	11.0
<i>max-elevation-turn</i>	Maximum turning angle that an agent can perform in order to fly along a controlled topographic contour.	3.7
<i>max-loft-turn</i>	Maximum turn angle that an agent can make in order to fly in a general bearing towards the home loft.	1.9
<i>max-separation-turn</i>	Maximum turning that an agent performs in order to be in separate from nearby flock mates.	3.5
<i>max-step-distance</i>	Maximum distance that an agent can traverse in a single time step.	
<i>minimum-separation-m</i>	Least distance that is necessary to avoid collision between pigeons in a flock.	1.25
<i>min-step-distance</i>	Least distance that an agent can traverse in a single time step in order to continue with navigation.	
<i>mutation-rate</i>	Rate at which mutation in candidate chromosomes occur.	0.01
<i>random-turn-prob</i>	Probability of an agent making a random turn.	0.3
<i>replace-proportion</i>	Proportion of candidate agents that are replaced in each time step.	0.6
<i>view-angle</i>	Maximum angular radius that is viewable to an agent at any instance.	340
<i>vision-m</i>	Maximum distance that is visible to an agent at any instance.	100

Apart from the mutation rate and replacement proportion parameters which were arbitrarily specified by the authors, the remaining initial model parameters in Table 3 were obtained by specifying and calibrating a conventional model. The purpose of the conventional model was to identify a suitable range of parameters for simulating flocking and navigation behaviour of homing pigeons [33]. Calibration was carried out against GPS tracks of first homing flight (homing flight 1) in the experimental data.

2.2.8. Initialization

At the beginning of the simulation, the number of birds in the empirical data, which is used as the basis of specifying the behaviour of “real” agent, is similarly used to specify the number of “simulated” agents to be created and simulated in the model. This was done to have the simulated agents as a mirror of their real (observed) counterparts. The first set of coordinates of the “real” agents was used to position the “simulated” agent in the model world. Each “simulated” agent was assigned a chromosome specifying the parameters for navigation. The chromosomes were encoded as linear vectors of the flocking and navigation parameters which included: maximum alignment turn (A_t), maximum coherence turn (C_t), maximum separation turn (S_t), maximum loft turn (L_t), maximum elevation turn (E_t), vision distance (V_d), maximum view angle (V_a), minimum step distance (S_n), maximum step distance (S_m), and random turn angle (R_t). Figure 2 is an illustration of how the model parameters are organized within a chromosome of an agent. This specification is actualized using list data structures in NetLogo.

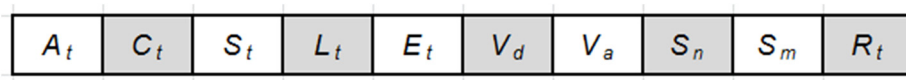


Figure 2. An illustration of the structure of a candidate chromosome. Constituent navigation and flocking parameters are represented as the genes of the chromosome.

A random disturbance is applied to the parameters by adding a random value from a normal distribution with a mean of 0.0 and standard deviation of 0.1 to each of the elements of a chromosome. This ensured that each agent is assigned a unique initial chromosome upon which to base its flocking and navigation decisions.

2.3. Parameter Estimation and Optimization

At each time step, a number of state variables are estimated to describe the state of different agents during the life of the model. State variable of interest in this task included, the step distance, the flight distance, relative turn angle, sinuosity of the flight path, and the fitness of candidate agents (in the training runs of the model). Step distance (S_{d_i}) is the discrete distance traversed by an agent in a single time step. This is estimated as a function of minimum step distance (S_{n_i}) and maximum step distance (S_{m_i}) as shown in Equation (2).

$$\text{Given } S_{n_i}, S_{m_i} \in \mathbb{R}, S_{d_i} = S_{n_i} + \text{rand} \quad (2)$$

where $\text{rand} \in [0, |S_{m_i} - S_{n_i}|]$.

Flight distance (S_{f_i}) is the cumulative length of the flight path traversed by an agent and is estimated as the sum of individual step distances from the beginning of the model to the current time step. Relative turn angle is estimated as the sum of the navigation, flocking, and random turns that are made by an agent at each time step. Sinuosity is a measure of the efficiency of the path that is followed by an agent and is calculated as the ratio of the flight distance and the Euclidean distance between the release sight and the current location of the agent. Fitness in the context of this work is a measure of how well the candidate navigation parameters simulate the flight paths of homing

pigeons and is computed as a function of sinuosity of the flight path and proximity of the simulated agent locations to the location of “real” agents in the model. Specifically, we estimated instantaneous fitness of the candidate parameters (f_i) as a sum of the component of fitness (f_d) resulting from the Euclidean distance (d) between the locations of “simulated” agents and the emulated GPS locations at a point in time and the component of fitness (f_s) resulting from the difference in sinuosity (s) of the paths of “simulated” agents and the paths of the “real” agents. Equations (3) and (4) show how the two components of fitness function are estimated. Our choice of sinuosity as one of the variables in the fitness function was motivated by the fact that sinuosity is a composite navigation parameter. Specifically, sinuosity encompasses the influence of random behaviour, relative turn angles and step-distances to the navigation behaviour of a moving agent. Additionally, sinuosity and locations of the simulated flight paths are functions of turn angles and step distances. As such, we considered a fitness function derived from a combination of sinuosity and positional differences of flight paths to be appropriate for testing the sensitivity of our model parameters during simulation.

$$f_d = \begin{cases} 1, & \text{if } d = 0 \\ d^{-1}, & \text{if } 0 < d \leq 1000 \\ -0.01d, & \text{if } d > 1000 \end{cases} \quad (3)$$

$$f_s = \begin{cases} 1, & s = 0 \\ 0.01s^{-1}, & 0 < s \leq 0.5 \\ 0.01(s - 1), & s > 0.5 \end{cases} \quad (4)$$

In the implementation of genetic algorithm, flocking and navigation decisions of “simulated” agents are guided by the parameters as specified in the elements of their respective chromosomes. The decisions made by simulated agents at each time step were evaluated by their associated fitness values which were calculated according to the fitness function as specified in Equations (3) and (4). Depending on a predefined replacement rate (*replace-proportion*), a proportion of simulated agents, characterised by low fitness values at the end of each time step were eliminated from the model. An elitist selection method was used to select two parent agents from the remaining proportion of the agents. The selected parents hatch offspring in place of the eliminated agents. Elitist selection operators which include ranking and tournament selection have been found to be more efficient when compared to proportional and roulette based selection schemes [70,71]. The chromosomes for the new generation of agents were created through a single-point crossover of the parent chromosomes. The crossover/recombination process was achieved with the help of *create-new-generation* procedure in the model. Additionally, a mutation procedure (*mutate*) was implemented to randomly vary the genes of chromosomes of the simulated agents. This process was executed iteratively for as long as there were emulated sensor data streams against which to evaluate the fitness of simulated agents. Fifty simulations were executed; in each run, chromosomes of agents in the final generation of agents and the associated fitness values were recorded. The parameters in these final set of chromosomes of each model run were considered as the optimized set of parameters from the respective model run and were subjected to further analysis in R statistical package. The goal of further statistical analysis was to identify the statistical distribution and patterns of optimised parameters.

We validated the results of the model against the emulated sensor data streams from homing flights 3, 4, and 5 of the empirical GPS tracks data. Specifically, validation was carried out against three state variables including relative turn angles, step distance, and sinuosity of flight paths. We used geometric means and standard deviation of the optimized parameters to generate parameters that were used in the validation runs of the model. Specifically, this involved executing a random normal function (*random-normal*) in NetLogo to generate respective validation parameters based on the estimated mean and standard deviation of optimised parameters. BehaviorSpace tool within NetLogo was used to iteratively execute the validation runs of the model while using the optimised range as the boundaries of the parameter space. We then used Student’s *t-test* at 95% confidence interval to

compare values of step distance (*step-distance*) and the relative turn angles (*mean-turn*) of simulated and the empirical flight tracks. Additionally, we plotted the sinuosity of the simulated tracks and the empirical tracks to provide a visual comparison in the characteristics of the flight paths of the simulated agents and the real agents. Results of this analysis are presented in the next section. Finally, we used coordinates of agents in the validation runs to map the flight paths of simulated agents and to estimate the tentative corridors emerging from the respective flight paths.

3. Results

An important output from this study is the set of optimized parameters upon which pigeon agents can navigate from a release site to a known loft. Table 4 represents a summary of the means and the confidence interval of optimized parameters of navigation of homing pigeons as implemented in this study. It should be noted that flocking is an important aspect of this model and so are the associated parameters as these are important to ensure that pigeons navigate in flocks. All angular parameters are in degrees ($^{\circ}$) while the distance related parameters are presented in meters (m). With the exception of the random turn angle ($\pm 2.5^{\circ}$), a majority of the angular parameters were characterized within the narrow width of the confidence intervals ($\pm 0.3^{\circ}$).

Table 4. Average of flocking and navigation parameters as calculated from the model.

Parameters	Mean (95% CI)
maximum alignment turn angle	8.01 (7.87–8.16)
maximum cohere turn angle	10.93 (10.81–11.04)
maximum separation turn angle	2.36 (2.22–2.51)
maximum compass turn angle	1.94 (1.82–2.06)
maximum elevation turn angle	3.63 (3.47–3.78)
vision distance	100.05 (99.9–100.2)
view angle	340 (339.8–340.1)
minimum step distance	2.45 (2.34–2.56)
maximum step distance	7.07 (6.8–7.35)
random turn angle	14.72 (12.29–17.16)

We further plotted density distribution charts of optimal range of navigation parameters (Figure 3) to provide a visual demonstration of the spread of the respective parameters. The mean value of the optimised parameters is indicated by the dotted lines. Once again, we observed that, in all cases, the optimized range of parameters were generally leptokurtic with a concentration of optimized values around the mean.

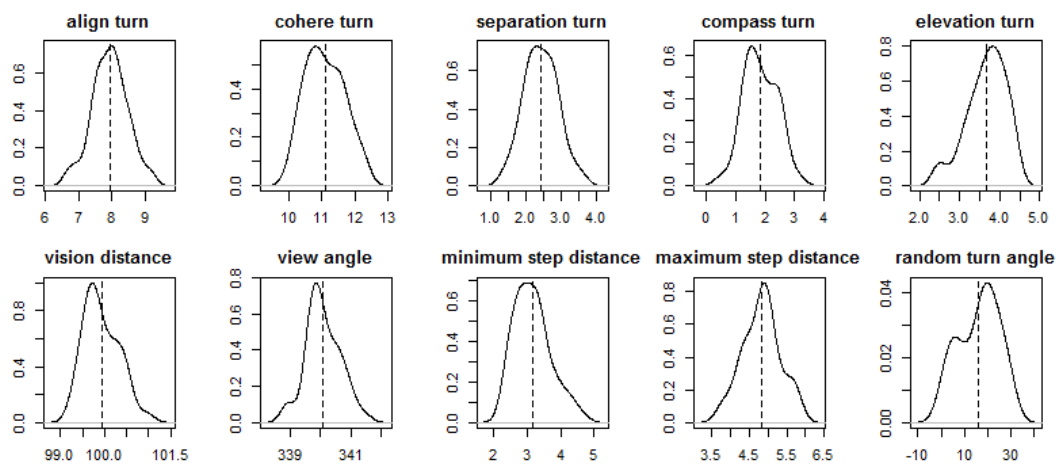


Figure 3. Density distribution of optimal model parameters.

In addition to the optimised parameters, we also plotted the temporal changes of fitness values of best performing parameters in each of the experimental model scenarios. The variation of fitness values against the model time (ticks) is shown in Figure 4. This graph was plotted from aggregated values of 50 exemplary simulation runs. We observed sharp increases in the fitness values of best performing chromosomes in the first 500 time steps (0–2 min), followed by moderate improvement of fitness in the period between 500–2600 time steps (2–10 min), and thereafter another period of rapid rise in the fitness values as agents inch closer to the homing loft.

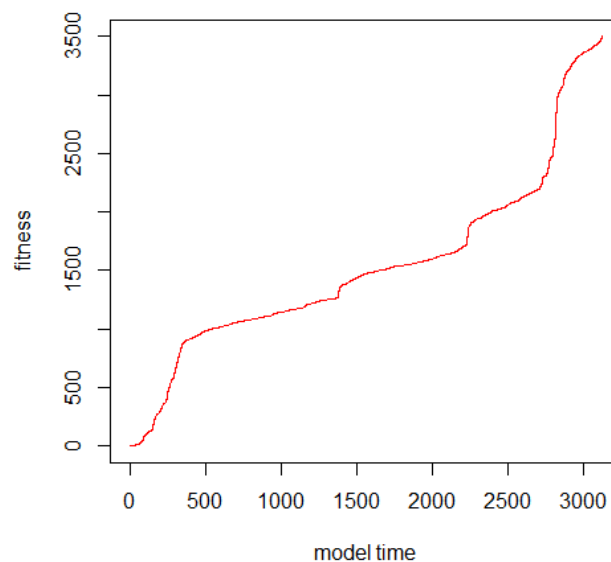


Figure 4. Variation of fitness of candidate parameters during model execution.

The optimized range of parameters provided the input for validation runs of the model. The output of validation runs included instantaneous coordinates of simulated agents, values of relative turn angles, step-distances, and estimated sinuosity of the flight path at each time step. Maps of simulated flight tracks against the validation data are presented in Figure 5. Apart from the simulated locations of birds that were captured in the validation runs, we used a kernel density estimation (KDE) surface to reveal the probable flight corridors in the area of study. Our choice of KDE as a method to estimate the flight corridor was motivated by the fact that the method has been used extensively [72,73] to estimate home range of different organisms. Additionally, we observed that, by using the optimized range of parameters in simulation, it was possible to replicate loops in flight paths. This is a pattern that was also evident in the empirical pigeon flight tracks; an example is highlighted in Figure 6.

A statistical comparison of the simulated state variables of step distance and relative turn angles was achieved in two parts; firstly, we used box and whisker plots to represent the median and the distribution of the observed and simulated variables as shown in Figure 7. We noted that whereas the median of the observed value of step distance was approximately 4.5 m, the median of the simulated step distance was slightly higher at approximately 4.7 m. The mean values of the observed and simulated values of relative turn angles during navigation were in both cases approximately equal to 0° . Additionally, for both the step distance and the relative turn angles, the range of simulated values was slightly wider in comparison to the range of the observed values (3 m to 6.4 m against 3.6 m to 5.3 m the case of step distances and -10° to 10° against -5° to 5° in the case of relative turn angles). Results of T-test showed that the mean values of step distance of pigeon agents as simulated from the optimised navigation parameters ($4.74 \text{ m} \pm 0.25 \text{ m}$) and those recorded in the empirical data ($4.47 \text{ m} \pm 0.27 \text{ m}$) were not statistically different (p -value = 0.012). Similarly, the mean value of the

simulated relative turn angle ($0.02^\circ \pm 0.02$) was statistically equal to the mean value from the observed data ($-0.07^\circ \pm 0.15$) with a p -value < 0.005 .

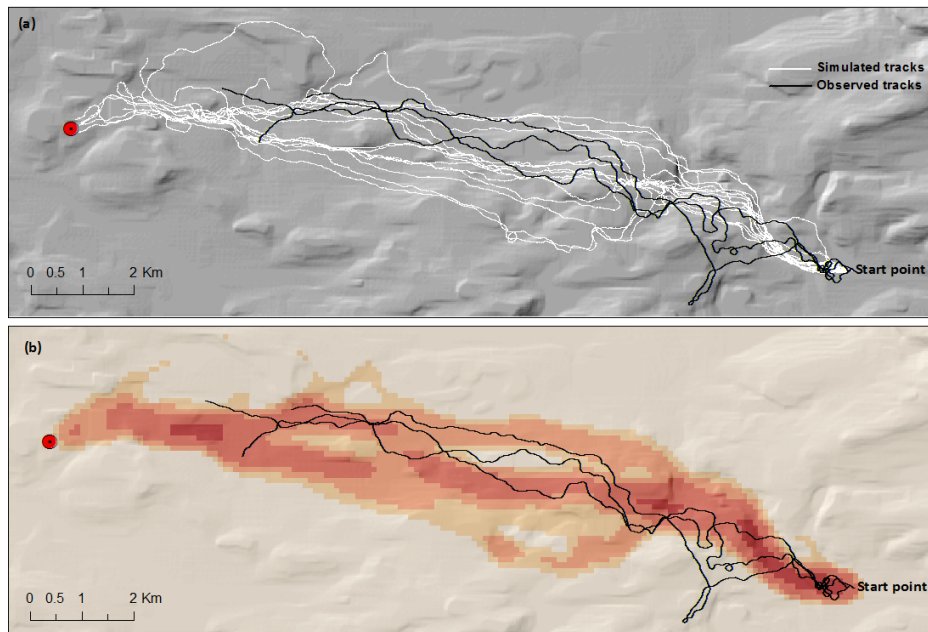


Figure 5. (a) Map of simulated and observed (validation) flight tracks of homing pigeons; (b) Map of validation flight tracks overlaid on a kernel density surface of locations of simulated tracks of homing pigeons.

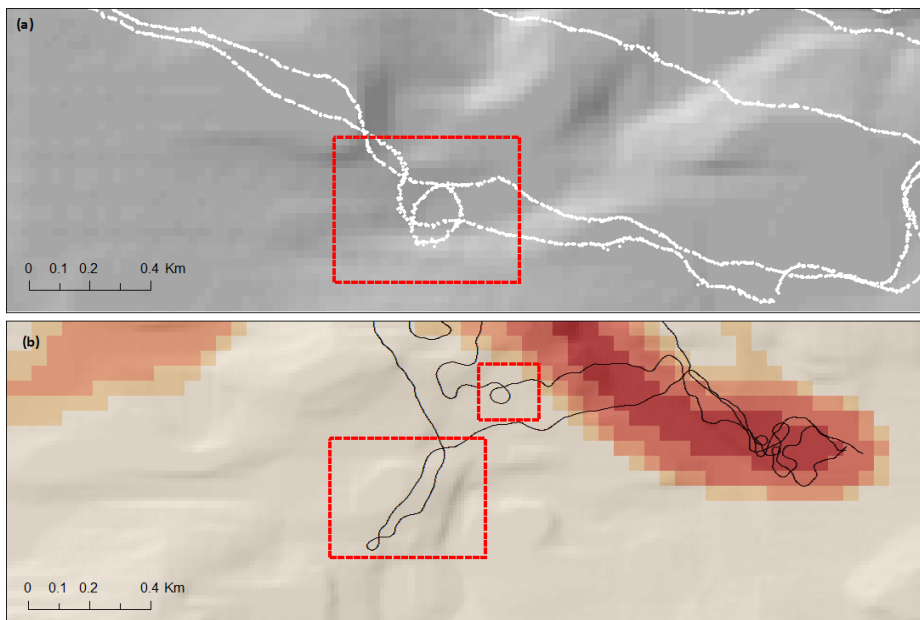


Figure 6. Loops in flight paths as reproduced by optimized model parameters (a); loops as captured in the empirical data (b).

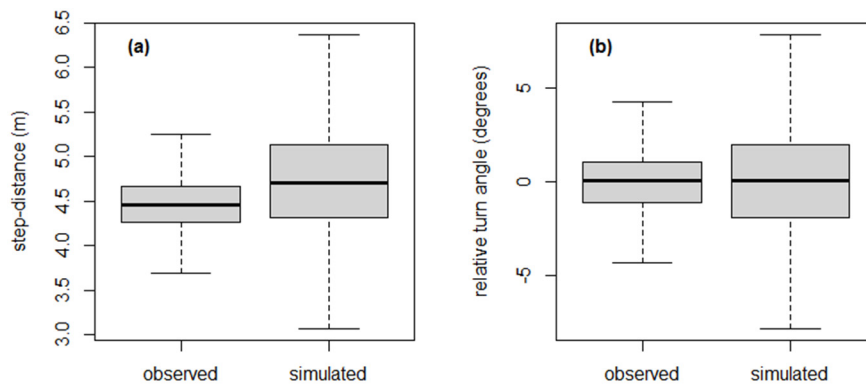


Figure 7. Box and whisker plots of observed and simulated state variables including, step-distance (a) and relative turn angles (b).

The final set of outputs that we considered from this study was the temporal variation in sinuosity of flight paths, a graphic representation of which is shown in Figure 8. In the results, we observed that the chart of sinuosity of the flight paths of real pigeons that we generated by aggregating individual values of birds in homing flights 3, 4, and 5 was generally above the simulated values indicating that the refined navigation parameters result in agents following slightly more efficient routes. Additionally, whereas real pigeons spend approximately the first two minutes (≈ 500 ticks) after release to re-orient, the optimized parameters did not take this into account, hence underestimating sinuosity of flight paths in the initial stages of the simulation.

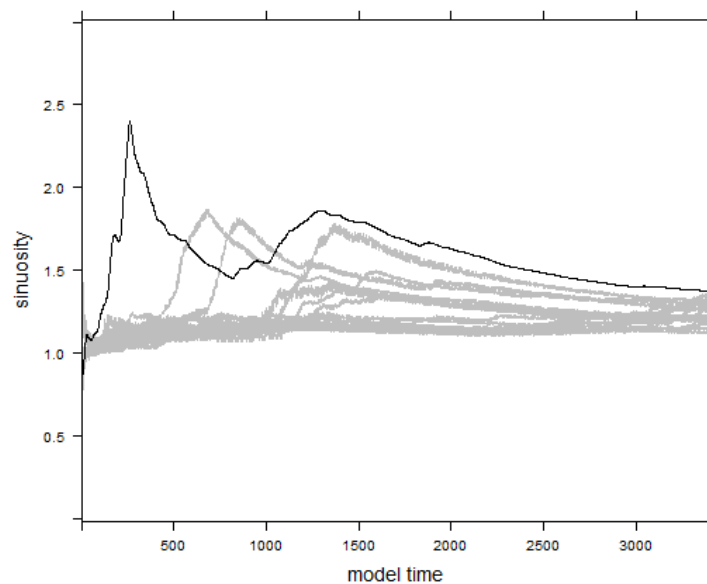


Figure 8. Variation of sinuosity of flight paths of homing pigeons as observed in the empirical data (black) and as simulated using optimized flight parameters (grey).

4. Discussion

In this study, we used emulated sensor data streams to provide a continuous mechanism of evaluating agent decisions during the life of a model. The result is a set of optimized model parameters that can be used to simulate the navigation behaviour of homing pigeons. In contrast to conventional models, whose output are stand-alone sets of calibrated parameters which can be considered as the most realistic representation of the system of interest, the implementation of genetic algorithms for optimisation resulted in a range of values for each parameter. This allowed for a wider scope of

parameters space on which to the model can be validated. Our model was particularly accurate in simulating key state variables in the model including step-distances and relative turn angles of agents. Optimisation also has the benefit of resulting in robust parameters that can replicate emergent patterns in the behaviours of agents. In this work, optimized parameters successfully replicated loops in the flight paths of homing pigeons, a pattern that has been observed in other studies [74]. This success was initially not possible by means of the conventional ABM paradigm [33]. Even though it is difficult to establish a mathematical relationship between the model parameters and patterns such as loops, such patterns do provide qualitative evidence that can be used in pattern-oriented modelling [75].

We observed that fitness values of the candidate model parameters improved with the progress of the models (Figure 4), indicating a refinement in the model parameters towards an optimal representation of the underlying system. While this underscores the suitability of evolutionary methods in modelling and simulating dynamic phenomena in data-rich environments, such methods also run the risk of creating “super-fit” solutions. This is one of the challenges that we faced in our model implementation. In particular, while our model captured the trend of changes in sinuosity of the flight path, the same were generally lower than the observed values. One suggestion to dealing with model overfitting when using genetic algorithms is to have a proper parental selection that limits the chance of superfit solutions from taking over the evolution process [76]. Additionally, other studies [77] have suggested the adoption of hybrid approaches that use conventional modelling methods to complement genetic algorithms by providing provisional calibrated parameters for the initial formulation candidate solution in genetic algorithms. This is the approach that we adopted in our implementation. As an alternative, initial parameters can also be obtained from well documented research and from domain experts [78]. Such an approach has the advantage of providing an anchor between existing knowledge and new research. As an example, hybridity attained by combining genetic algorithms and local optimisers has been used to improve the calibration of cellular automata models [79].

Our findings are in agreement with other studies that have found that data assimilation into ABMs can improve the predictive ability of such models by facilitating agent behaviour specification [80] and model calibration [81]. Additionally, by choosing a fitness function that is sensitive to the various model parameters [82], the use of genetic algorithms made it possible to simultaneously carry out optimization and sensitivity analysis [45]. Choosing a method that incorporates sensitivity analysis as part of optimization improves the efficiency of the models. The methods that we have presented in this paper contribute towards finding alternatives to traditional simulation models. Specifically, conventional rule-based models have been characterized by rigid parameter settings and minimal use of the real time data from the systems under study [23]. We handled these challenges by using genetic algorithms to improve the parameters of the model during the simulation. Additionally, the methods allowed for an efficient and dynamic integration of spatio-temporal data into the model, thus improving the predictive ability of the model.

In this study we did not keep a record of the memory of the individual agents, such an attempt would be of interest especially as a way of investigating the actual triggers of adaptation and the influence of local spatial characteristics to the learning and evolution process. Additionally, we did not consider the influence of changes of the environment on the behaviour of agents. We see this as an opportunity for future research. In particular, the integration of in-situ sensor observations and dynamic environmental variables (including vegetation indices and climatic variables) from remote sensing into ABMs should be considered. Additionally, we foresee additional opportunities particularly in the integration of dynamic data assimilation [81] methods and evolutionary algorithms in the implementation of self-calibrating and adaptive agent based models. This will be particularly useful to take advantage of advanced sensor data capture methods and associated geosensor networks [83]. Additionally, animal tracking and data management endeavours like the upcoming International Cooperation for Animal Research Using Space (ICARUS) initiative will also provide a good resource for researchers with an interest in dynamic data assimilation in behavioural models of animal movement.

The aim of ICARUS project is to observe/capture global migratory patterns of small animals and to allow scientists to gain insight into vital functions and behaviour of animals [84]. We believe that the knowledge from our study will be useful in implementing automated agent behaviour detection models against real geospatial resources as those that will be captured by the ICARUS project.

5. Conclusions

In this study, we set out to investigate on how spatio-temporal data about a real world system can be used to improve the specification of a rule-based ABM. We evaluated this objective by implementing an evolutionary algorithm against data depicting the navigation of homing pigeons. Specifically, emulated sensor data from GPS tracks provided a dynamic spatial and temporal representation of the behaviour of autonomous agents. Consequently, these availed a basis of evaluating simulated agent behaviour and thus giving credence to the hypothesis that dynamic data streams from sensor observations can be incorporated into agent-based models to improve the formulation of agent behaviours and facilitate the understanding of underlying systems. Our model showed that, even though agents as specified in this study were not trained to remember all the possible states of their environment, the optimised parameters were able to successfully simulate the core state variables of the model. In particular, the optimised parameters accurately simulated relative turn-angle and step-distance, which are vital for describing the trajectories of animal movement. Apart from predicting the state variables, the optimised parameters also replicated loops in the flight paths of homing pigeons. Loops are a distinctive pattern that are commonly observed in the flight paths of pigeons but have not been easy to replicate using conventional ABMs. It is evident that using fine-grained spatio-temporal data about the systems of interest to provide dynamic evaluation and optimization of the model parameters results in robust parameters. Such data-driven paradigms can improve the transferability of model parameters to other scenarios. More fundamentally, availability of dynamic and high resolution data brings into question the possibility of using such data to develop plausible models without overreliance on theories. This, we foresee as the next frontier for researchers with an interest in rule-based agent-based models.

Acknowledgments: This research is supported by the Austrian Science Fund (FWF) through the Doctoral College GIScience at the University of Salzburg (DK W1237-N23).

Author Contributions: Francis Oloo and Gudrun Wallentin conceived and designed the general frame of the model; Francis Oloo implemented the model and analysed the data; and Francis Oloo and Gudrun Wallentin wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. An, G.; Mi, Q.; Dutta-Moscato, J.; Vodovotz, Y. Agent-based models in translational systems biology. *Wiley Interdiscip. Rev. Syst. Biol. Med.* **2009**, *1*, 159–171. [[CrossRef](#)] [[PubMed](#)]
2. Xiang, X.; Kennedy, R.; Madey, G.; Cabaniss, S. Verification and validation of agent-based scientific simulation models. In Proceedings of the Agent-Directed Simulation Conference, New Orleans, LA, USA, 23–28 July 2005.
3. Heppenstall, A.; Malleson, N.; Crooks, A. “Space, the final frontier”: How good are agent-based models at simulating individuals and space in cities? *Systems* **2016**, *4*, 9. [[CrossRef](#)]
4. Gong, J.; Geng, J.; Chen, Z. Real-time GIS data model and sensor web service platform for environmental data management. *Int. J. Health Geogr.* **2015**, *14*, 2. [[CrossRef](#)] [[PubMed](#)]
5. Granell, C.; Díaz, L.; Gould, M. Service-oriented applications for environmental models: Reusable geospatial services. *Environ. Model. Softw.* **2010**, *25*, 182–198. [[CrossRef](#)]
6. Mineter, M.J.; Jarvis, C.; Dowers, S. From stand-alone programs towards grid-aware services and components: A case study in agricultural modelling with interpolated climate data. *Environ. Model. Softw.* **2003**, *18*, 379–391. [[CrossRef](#)]

7. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New generation sensor web enablement. *Sensors* **2011**, *11*, 2652–2699. [[CrossRef](#)] [[PubMed](#)]
8. Broering, A.; Foerster, T.; Jirka, S.; Priess, C. Sensor bus: an intermediary layer for linking geosensors and the sensor web. In Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, Washington, DC, USA, 21–23 June 2010.
9. Song, X.; Wang, C.; Kagawa, M.; Raghavan, V. Real-time monitoring portal for urban environment using sensor web technology. In Proceedings of the 2010 18th International Conference on the Geoinformatics, Beijing, China, 18–20 June 2010.
10. Resch, B.; Mittlboeck, M.; Girardin, F.; Britter, R.; Ratti, C. Real-time Geo-awareness–Sensor Data Integration for Environmental Monitoring in the City. In Proceedings of the International Conference on Advanced Geographic Information Systems & Web Services, Cancun, Mexico, 1–7 February 2009.
11. Resch, B.; Mittleboeck, M.; Lipson, S.; Welsh, M.; Bers, J.; Britter, R.; Ratti, C.; Blaschke, T. *Integrated Urban Sensing: A Geo-Sensor Network for Public Health Monitoring and Beyond*; MIT Press: Cambridge, MA, USA, 2011.
12. Sagl, G.; Blaschke, T.; Beinat, E.; Resch, B. Ubiquitous geo-sensing for context-aware analysis: Exploring relationships between environmental and human dynamics. *Sensors* **2012**, *12*, 9800–9822. [[CrossRef](#)] [[PubMed](#)]
13. Klug, H.; Kmoch, A. Operationalizing environmental indicators for real time multi-purpose decision making and action support. *Ecol. Model.* **2015**, *295*, 66–74. [[CrossRef](#)]
14. Laube, P.; Duckham, M.; Wolle, T. Decentralized movement pattern detection amongst mobile geosensor nodes. In Proceedings of the International Conference on Geographic Information Science, Park City, UT, USA, 23–26 September 2008; Springer: Berlin/Heidelberg, Germany, 2008.
15. Wikelski, M.; Kays, R. Movebank: Archive, Analysis and Sharing of Animal Movement Data. Available online: <http://www.movebank.org> (accessed on 17 January 2017).
16. Kranstauber, B.; Cameron, A.; Weinzerl, R.; Fountain, T.; Tilak, S.; Wikelski, M.; Kays, R. The Movebank data model for animal tracking. *Environ. Model. Softw.* **2011**, *26*, 834–835. [[CrossRef](#)]
17. Dettki, H.; Brode, M.; Clegg, I.; Giles, T.; Hallgren, J. Wireless remote animal monitoring (WRAM)—A new international database e-infrastructure for management and sharing of telemetry sensor data from fish and wildlife. In Proceedings of the 7th International Congress on Environmental Modelling and Software, San Diego, CA, USA, 15–19 June 2014.
18. Cagnacci, F.; Urbano, F. Managing wildlife: A spatial information system for GPS collars data. *Environ. Model. Softw.* **2008**, *23*, 957–959. [[CrossRef](#)]
19. Urbano, F.; Cagnacci, F.; Calenge, C.; Dettki, H.; Cameron, A.; Neteler, M. Wildlife tracking data management: A new vision. *Philos. Trans. R. Soc. B Biol. Sci.* **2010**, *365*, 2177–2185. [[CrossRef](#)] [[PubMed](#)]
20. Duckham, M.; Reitsma, F. Decentralized environmental simulation and feedback in robust geosensor networks. *Comput. Environ. Urban Syst.* **2009**, *33*, 256–268. [[CrossRef](#)]
21. Duckham, M.; Yeoman, J. Decentralized Algorithms and Simulations for Detecting and Monitoring Convoys. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.716.7594&rep=rep1&type=pdf> (accessed on 17 January 2017).
22. Darema, F. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In Proceedings of the Computational Science—ICCS 2004, Kraków, Poland, 6–9 June 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 662–669.
23. Hu, X. Dynamic data driven simulation. *SCS M&S Mag.* **2011**, *1*, 16–22.
24. Akçelik, V.; Biros, G.; Draganescu, A.; Hill, J.; Ghattas, O.; Waanders, B.V.B. Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants. In Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, Seattle, WA, USA, 12–18 November 2005.
25. Gaynor, M.; Seltzer, M.; Moulton, S.; Freedman, J. A dynamic, data-driven, decision support system for emergency medical services. In Proceedings of the Computational Science—ICCS 2005, Atlanta, GA, USA, 22–25 May 2005.
26. Celik, N.; Lee, S.; Vasudevan, K.; Son, Y.-J. DDDAS-based multi-fidelity simulation framework for supply chain systems. *IIE Trans.* **2010**, *42*, 325–341. [[CrossRef](#)]
27. Pereira, G.M. Dynamic data driven multi-agent simulation. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Fremont, CA, USA, 2–5 November 2007.

28. Li, Z.; Guan, X.; Li, R.; Wu, H. 4D-SAS: A Distributed Dynamic-Data Driven Simulation and Analysis System for Massive Spatial Agent-Based Modeling. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 42. [[CrossRef](#)]
29. Rai, S.; Hu, X. Behavior pattern detection for data assimilation in agent-based simulation of smart environments. In Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Atlanta, GA, USA, 17–20 November 2013.
30. Taormina, R.; Chau, K.-W. Data-driven input variable selection for rainfall–runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines. *J. Hydrol.* **2015**, *529*, 1617–1632. [[CrossRef](#)]
31. Wu, C.L.; Chau, K.W.; Li, Y.S. Methods to improve neural network performance in daily flows prediction. *J. Hydrol.* **2009**, *372*, 80–93. [[CrossRef](#)]
32. Zhang, H.; Vorobeychik, Y.; Letchford, J.; Lakkaraju, K. Data-driven agent-based modeling, with application to rooftop solar adoption. *Auton. Agents Multi-Agent Syst.* **2016**, *30*, 1023–1049. [[CrossRef](#)]
33. Wallentin, G.; Oloo, F. A Model-sensor Framework to Predict Homing Pigeon Flights in Real Time. *GI_Forum* **2016**, *1*, 41–52. [[CrossRef](#)]
34. McLane, A.J.; Semeniuk, C.; McDermid, G.J.; Marceau, D.J. The role of agent-based models in wildlife ecology and management. *Ecol. Model.* **2011**, *222*, 1544–1556. [[CrossRef](#)]
35. Bennett, D.A.; Tang, W. Modelling adaptive, spatially aware, and mobile agents: Elk migration in Yellowstone. *Int. J. Geogr. Inf. Sci.* **2006**, *20*, 1039–1066. [[CrossRef](#)]
36. Hamblin, S.; Giraldeau, L.-A. Finding the evolutionarily stable learning rule for frequency-dependent foraging. *Anim. Behav.* **2009**, *78*, 1343–1350. [[CrossRef](#)]
37. Hamblin, S. On the practical usage of genetic algorithms in ecology and evolution. *Methods Ecol. Evol.* **2013**, *4*, 184–194. [[CrossRef](#)]
38. Okunishi, T.; Yamanaka, Y.; Ito, S.-I. A simulation model for Japanese sardine (*Sardinops melanostictus*) migrations in the western North Pacific. *Ecol. Model.* **2009**, *220*, 462–479. [[CrossRef](#)]
39. Kawecki, T.J.; Ebert, D. Conceptual issues in local adaptation. *Ecol. Lett.* **2004**, *7*, 1225–1241. [[CrossRef](#)]
40. Wechsler, B.; Lea, S.E. Adaptation by learning: Its significance for farm animal husbandry. *Appl. Anim. Behav. Sci.* **2007**, *108*, 197–214. [[CrossRef](#)]
41. Lauer, M.; Riedmiller, M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000.
42. Macal, C.M.; North, M.J. Tutorial on agent-based modelling and simulation. *J. Simul.* **2010**, *4*, 151–162. [[CrossRef](#)]
43. Stonedahl, F.; Wilensky, U. Finding forms of flocking: Evolutionary search in abm parameter-spaces. In Proceedings of the International Workshop on Multi-Agent Systems and Agent-Based Simulation, Toronto, ON, Canada, 11 May 2010; Springer: Berlin/Heidelberg, Germany, 2010.
44. Stonedahl, F.; Rand, W.M. *When Does Simulated Data Match Real Data? Comparing Model Calibration Functions Using Genetic Algorithms*; Robert H. Smith School Research Paper No. RHS-06-151; University of Maryland: College Park, MD, USA, 2012.
45. Stonedahl, F.; Wilensky, U. Evolutionary robustness checking in the artificial anasazi model. In Proceedings of the AAAI Fall Symposium: Complex Adaptive Systems, Arlington, VA, USA, 11–13 November 2010.
46. Holland, J.H. *Complex Adaptive Systems*; Daedalus; The MIT Press: Cambridge, MA, USA, 1992.
47. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
48. Mitchell, M. Genetic algorithms: An overview. *Complexity* **1995**, *1*, 31–39. [[CrossRef](#)]
49. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.
50. Calvez, B.; Hutzler, G. Automatic tuning of agent-based models using genetic algorithms. In Proceedings of the International Workshop on Multi-Agent Systems and Agent-Based Simulation, Paris, France, 4–6 July 2005; Springer: Berlin/Heidelberg, Germany, 2005.
51. Barta, Z.; Flynn, R.; Giraldeau, L.-A. Geometry for a selfish foraging group: A genetic algorithm approach. *Proc. R. Soc. Lond. B Biol. Sci.* **1997**, *264*, 1233–1238. [[CrossRef](#)]
52. Beauchamp, G. A spatial model of producing and scrounging. *Anim. Behav.* **2008**, *76*, 1935–1942. [[CrossRef](#)]
53. Oremland, M.; Laubenbacher, R. Optimization of agent-based models: Scaling methods and heuristic algorithms. *J. Artif. Soc. Soc. Simul.* **2014**, *17*, 6. [[CrossRef](#)]

54. Mandel, J.; Beezley, J.D.; Bennethum, L.S.; Chakraborty, S.; Coen, J.L.; Douglas, C.C.; Hatcher, J.; Kim, M.; Vodacek, A. A dynamic data driven wildland fire model. In Proceedings of the International Conference on Computational Science, Beijing, China, 27–30 March 2007; Springer: Berlin/Heidelberg, Germany, 2007.
55. Rodríguez, R.; Cortés, A.; Margalef, T. Injecting dynamic real-time data into a DDDAS for forest fire behavior prediction. In Proceedings of the International Conference on Computational Science, Vancouver, BC, Canada, 29–31 August 2009; Springer: Berlin/Heidelberg, Germany, 2009.
56. Denham, M.; Cortés, A.; Margalef, T.; Luque, E. Applying a Dynamic Data Driven Genetic Algorithm to Improve Forest Fire Spread Prediction. In Proceedings of the 8th International Conference on Computational Science, Kraków, Poland, 23–25 June 2008; Bubak, M., van Albada, G.D., Dongarra, J., Sloat, P.M.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 36–45.
57. Lee, K.H.; Choi, M.G.; Hong, Q.; Lee, J. Group behavior from video: A data-driven approach to crowd simulation. In Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, CA, USA, 3–4 August 2007.
58. Nagy, M.; Akos, Z.; Biro, D.; Vicsek, T. Hierarchical group dynamics in pigeon flocks. *Nature* **2010**, *464*, 890–893. [[CrossRef](#)] [[PubMed](#)]
59. Blaser, N.; Dell’Omo, G.; Dell’Ariccia, G.; Wolfer, D.P.; Lipp, H.P. Testing cognitive navigation in unknown territories: Homing pigeons choose different targets. *J. Exp. Biol.* **2013**, *216*, 3123–3131. [[CrossRef](#)] [[PubMed](#)]
60. Santos, C.D.; Neupert, S.; Lipp, H.-P.; Wikelski, M.; Dechmann, D.K. Temporal and contextual consistency of leadership in homing pigeon flocks. *PLoS ONE* **2014**, *9*, e102771. [[CrossRef](#)] [[PubMed](#)]
61. Santos, C.D.; Neupert, S.; Lipp, H.-P.; Wikelski, M.; Dechmann, D. Data from: Temporal and Contextual Consistency of Leadership in Homing Pigeon Flocks. Available online: <https://www.datarepository.movebank.org/handle/10255/move.365> (accessed on 10 October 2016).
62. Grimm, V.; Berger, U.; DeAngelis, D.L.; Polhill, J.G.; Giske, J.; Railsback, S.F. The ODD protocol: A review and first update. *Ecol. Model.* **2010**, *221*, 2760–2768. [[CrossRef](#)]
63. Wilensky, U. *NetLogo (and NetLogo User Manual)*; Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL, USA, 1999.
64. Arundel, J.; Oldroyd, B.P.; Winter, S. Modelling honey bee queen mating as a measure of feral colony density. *Ecol. Model.* **2012**, *247*, 48–57. [[CrossRef](#)]
65. Polhill, J.G.; Parker, D.; Brown, D.; Grimm, V. Using the ODD protocol for describing three agent-based social simulation models of land-use change. *J. Artif. Soc. Soc. Simul.* **2008**, *11*, 3.
66. Tang, W.; Bennett, D.A. Agent-based modeling of animal movement: A review. *Geogr. Compass* **2010**, *4*, 682–700. [[CrossRef](#)]
67. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*; ACM: New York, NY, USA, 1987.
68. Wilensky, U. *NetLogo Flocking Model*; Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL, USA, 1998.
69. Mitchell, M.T.; Wilensky, U. *NetLogo Robby the Robot Model*; Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL, USA, 2012.
70. Noraini, M.R.; Geraghty, J. Genetic algorithm performance with different selection strategies in solving TSP. In Proceedings of the World Congress on Engineering, London, UK, 6–8 July 2011.
71. Goldberg, D.E.; Deb, K. A comparative analysis of selection schemes used in genetic algorithms. *Found. Genet. Algorithms* **1991**, *1*, 69–93.
72. Walter, W.D.; Fischer, J.W.; Baruch-Mordo, S.; VerCauteren, K.C. *What Is the Proper Method to Delineate Home Range of An Animal Using Today’s Advanced GPS Telemetry Systems: The Initial Step*; InTech: Rijeka, Croatia, 2011.
73. Downs, J.; Horner, M. Network-based kernel density estimation for home range analysis. In Proceedings of the Ninth International Conference on Geocomputation, Maynooth, Ireland, 3–5 September 2007.
74. Dell’Ariccia, G.; Dell’Omo, G.; Wolfer, D.P.; Lipp, H.-P. Flock flying improves pigeons’ homing: GPS track analysis of individual flyers versus small groups. *Anim. Behav.* **2008**, *76*, 1165–1172. [[CrossRef](#)]
75. Grimm, V.; Revilla, E.; Berger, U.; Jeltsch, F.; Mooij, W.M.; Railsback, S.F.; Thulke, H.-H.; Weiner, J.; Wiegand, T.; DeAngelis, D.L. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science* **2005**, *310*, 987–991. [[CrossRef](#)] [[PubMed](#)]

76. Beasley, D.; Martin, R.; Bull, D. An overview of genetic algorithms: Part 1. Fundamentals. *Univ. Comput.* **1993**, *15*, 58.
77. El-Mihoub, T.A.; Hopgood, A.A.; Nolle, L.; Battersby, A. Hybrid Genetic Algorithms: A Review. *Eng. Lett.* **2006**, *13*, 124–137.
78. Wendt, K.; Cortés, A.; Margalef, T. Knowledge-guided genetic algorithm for input parameter optimisation in environmental modelling. *Procedia Comput. Sci.* **2010**, *1*, 1367–1375. [[CrossRef](#)]
79. Whitsed, R.; Smallbone, L.T. A hybrid genetic algorithm with local optimiser improves calibration of a vegetation change cellular automata model. *Int. J. Geogr. Inf. Sci.* **2016**, 1–21. [[CrossRef](#)]
80. Sajjad, M.; Singh, K.; Paik, E.; Ahn, C.-W. A Data-Driven Approach for Agent-Based Modeling: Simulating the Dynamics of Family Formation. *J. Artif. Soc. Soc. Simul.* **2016**, *19*, 1–14. [[CrossRef](#)]
81. Ward, J.A.; Evans, A.J.; Malleson, N.S. Dynamic calibration of agent-based models using data assimilation. *R. Soc. Open Sci.* **2016**, *3*, 150703. [[CrossRef](#)] [[PubMed](#)]
82. Lee, J.-S.; Filatova, T.; Ligmann-Zielinska, A.; Hassani-Mahmoei, B.; Stonedahl, F.; Lorscheid, I.; Voinov, A.; Polhill, J.G.; Sun, Z.; Parker, D.C. The Complexities of Agent-Based Modeling Output Analysis. *J. Artif. Soc. Soc. Simul.* **2015**, *18*. [[CrossRef](#)]
83. Nittel, S. A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring. *Sensors* **2009**, *9*, 5664–5678. [[CrossRef](#)] [[PubMed](#)]
84. Gross, M. Animal moves reveal bigger picture. *Curr. Biol.* **2015**, *25*, R585–R588. [[CrossRef](#)] [[PubMed](#)]



© 2017 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).